

M E M O R A N D U M

Date: July 21, 1998 (*updated*)
July 10, 1998

To: PCMDI Staff

From: PCMDI Computation Staff

Subject: Software Development and Integration Plan Requirements

CDAT

CDAT was developed as PCMDI's future software development foundation. Its purpose is to provide atmospheric scientists with easy and fast methods to ingest, analyze and store climate data. Its original requirements were that it use a standard mathematical library, have graphics capabilities, have a robust scripting language, allow access via an application programmer's interface, is application and user extensible, and have World-Wide Web access.

1) Assigned to Susan:

CDAT Documentation – the documentation must be on-line as a web document and the CDAT Help GUI must be up-to-date.

Person power required: 2 weeks

2) Assigned to Dean:

Adding Missing Value to All Numeric Functions – Currently, missing_value only works for 11 Numeric functions. In addition, missing value should work with ArrayObject functions (e.g., a.atype – when the data type changes, then the missing_value's data type should change as well.)

Person power required: 3 weeks

3) Assigned to Clyde:

Regridder – The regridder needs the functionality of EzGet. In particular, it needs regional masking and full control over the domain limits. In addition, it should handle "Unknown" grids including wrapping in the longitude and latitude direction.

Person power required: 2 weeks

4) Assigned to Dean:

Run VCS from the Command Line – The user must be able run VCS from the command line interface (i.e., Xterm window).

Person power required: 1 week

5) Assigned to Everyone (fix bugs as they occur in your software area):

CDAT Bug Fixes – bugs must be fixed as they occur. Currently, there are no outstanding bug fixes.

Person power required: continuous process

6) CDAT's Official Release:

CDAT will be ready for release Version 1.0 when action item 1 (CDAT Documentation), action item 2 (Adding Missing Value to All Numeric Functions), action item 3 (Regridding), action item 4 (Run VCS from the Command Line), and action item 5 (CDAT Bug Fixes) are completed.

CDAT will be released in August on the following platforms: Sun/Solaris 2.5 or higher, SGI/IRIX 6.0 or higher, and PC/Linux 5.0 or higher.

Note: the PC machine named Aric, located in Dean's office, is designated as the office PC/Linux platform for compiling CDAT.

7) Assigned to Susan:

FORTTRAN Interface - There must be a way to allow climate scientist access to their FORTRAN subroutines. This access should be clean and easy to use. Although, there may be some work on the part of the climate scientist to get their routines as clean, efficient and self-contained as possible. (This topic warrants a computer scientist group discussion on the best way to add a FORTRAN interface to CDAT's environment.)

Person power required: ???

8) Assigned to Susan:

FORTTRAN I/O - CDAT must be able to read and write FORTRAN I/O. Jim knows of a Python FORTRAN I/O module.

Person power required: ???

9) Assigned to Dean:

VCS - CDAT and VCS must also be fully compatible. More VCS access must be made for the CDAT user (i.e., manipulation of the template, graphics methods, and data). VCS must be modified to accommodate CDAT needs.

This action item may depend on whether or not the redesign VCS is done first.
(See the VCS section for further details.)

Person power required: ???

10) Assigned to Dean:

VCS Scripts - CDAT must be able to run all of VCS scripts from its interface. Currently, CDAT can run 90% of VCS scripts.

This action item must wait on the redesign VCS scripting language. We must first replace VCS's scripting language with Python's scripting language.
(See the VCS section for further details.)

Person power required: ???

11) Assigned to Susan:

Ben Santer's Functions - With the FORTRAN Interface (this may be trivial).

Person power required: 2 weeks

12) Assigned to Everyone:

Error Output – Must have consistent error output messages throughout PCMDI modules. Also, Dean is responsible for redirecting standard error and standard output messages to the CDAT Output window.

Person power required: continuous process

13) Assigned to Everyone:

Integrating Climate Scientist's Modules - We need to integrate the user's generated modules into CDAT.

Person power required: continuous process

14) Assigned to Everyone:

Porting to Additional Platforms – Everyone is responsible for porting their portion of CDAT to the various platforms.

Person power required: continuous process

15) Assigned to Everyone:

Upgrading CDAT to Python 1.5 – Everyone is responsible for making sure their modules work in the Python 1.5 environment.

Person power required: 1 week

16) Assigned to Susan:

Incorporate the Python Debugger into CDAT

Person power required: 3days

CDMS - Includes cdunif and PSQL

CDMS is PCMDI's data management system. It provides access to PCMDI data interface which support several levels of abstraction, including: the netCDF-like application programmer's interface (i.e., cdunif), file-oriented views and inquiry of data (i.e., PSQL), and abstract, powerful, and logical views of datasets, variables, axes, and grids (i.e., CDMS – the database itself).

1) Assigned to Charlie:

Spanning Files - Group or Spanning files must be made easier to use for the atmospheric scientists. (This topic warrants a PCMDI staff discussion on what it is the user would like to see in the spanning of files.)

Person power required: ???

2) Assigned to Charlie:

PSQL/CDAT – Processing AMIP II data by inquiring information about the data, identifying bad files, and re-writing the data as LATS generated netCDF files.

Eventually, this processing will include Quality Control Software (QCS).

Person power required: ???

3) Assigned to Bob:

API - The CDMS application programmer's interface must be finished in order to complete VCS module and CDAT. At this point, we are going to implement the API in Python (i.e., as discussed in meetings).

Person power required: ???

4) Assigned to Bob and Charlie:

CDMS Documentation - Documentation must be completed and put on the Web for users.

Person power required: ???

5) Assigned to Bob:

Cdunif FORTRAN Interface – FORTRAN application interface for cdunif is needed.

Person power required: ???

6) Assigned to Bob:

Cdunif Documentation - Documentation must be completed and put on the Web for users.

Person power required: ???

7) Assigned to Bob:

Cdunif/VPOP - LANL's ocean data format. Cdunif must be able to read this format in order to include this group into our software community.

Person power required: ???

8) Assigned to Charlie:

PSQL/Python Commands - There will need to be canned PSQL commands to assist users.

Person power required: ???

9) Assigned to Charlie:

PSQL Documentation - Documentation must be completed and put on the Web for users.

Person power required: ???

10) Assigned to Bob:

CDMS C and FORTRAN Interface - On completion of the Python application programmer's interface, a C and FORTRAN interface will be needed to give code developers access to CDMS function calls.

Person power required: ???

11) Assigned to Bob:

CDMS Graphical User's Interface – Must allow the user to edit the database via a GUI.

(This area warrants a group discussion.)

Person power required: ???

LATS

LATS is a library of software routines to output gridded data in the COARDS and GDT convention.

1) Assigned to Bob:

GDT - Add support for the GDT metadata conventions. Test and debug.

Person power required: completed

2) Assigned to Bob:

More Capabilities - Develop more general purpose output capabilities. (This topic warrants further discussion.)

Person power required: as needed

3) Assigned to Bob:

LATS Documentation - Documentation must be completed and put on the Web for users.

Person power required: continuous process

QCS

The Quality Control Software (QCS) will provide a semiautomatic procedure for quality control of AMIP II, CMIP, and PMIP model output data. The data will first be read and organized for evaluation of its consistency with respect to its integrity of time sequence. Hopefully, the data read is in LATS format. If not, then the data will be ingested in the best possible way or returned to its source. The evaluated/tested data will be rewritten in a standard time series using LATS to facilitate the ease in readability and distribution.

1) Assigned to Charlie:

Database Development - CDMS must work with accessing/inquiring the files.

Person power required: currently under development

2) Assigned to Charlie and Clyde:

First Order Quality Control (QC) Routines - QC routines must be developed to check for the integrity of the model data. This is currently being done with PSQL.

Person power required: currently under development

3) Assigned to Clyde and Charlie:

Processing Non-LATS Data Files – We may receive files that are not in LATS convention or netCDF format.

Person power required: continuous process

4) Assigned to Charlie and Clyde:

Second Order QC Routines - How does Karl's and Charles' quality control software fit into the QCS.

Person power required: ???

5) Assigned to Charlie/Dean/Peter:

Process Procedure - Develop a step or procedural process to the QC generation of model data.

Person power required: currently under development

VCS

The Visualization and Computation System (VCS) was designed to meet the visualization and graphic needs of the climate research scientists. It allows wide-range changes to be made to the data and plot(s) displayed on the VCS Canvas. In addition, it also provides hard-copy output, scripting capabilities, some calculations, and animation.

To meet the needs of our current and future PCMDI software environment it is necessary to change our view of VCS and restructure its code design according to the Python environment. It is mandatory to change VCS internal structure for several very important reasons:

Compatibility with Current PCMDI Software (i.e., CDAT) – Users have expressed their interest in VCS running CDAT/Python scripts from its point and click interface. In addition, they would like to use the calculation capability of CDAT in VCS's, thus, replacing VCS's computation section. By doing this, the user will be able to treat computed variables the same as any other variable retrieved from a data file.

Compatibility with PCMDI Current Internal Data Structure (i.e., cdunif) – VCS was designed with DRS as its internal data structure. This data structure is limited in several areas, including: the fixed number of variable attributes (i.e., source, name, title, units, date, time, etc.); the limited number of dimension attributes, currently allowing only four dimensions; and the limited number of data type, (i.e., integers, and floats). Since the design of VCS, PCMDI has developed and adopted the cdunif data model as its official ingest application programmer's interface.

Cdunif resolves the problems mentioned above and is fully compatible with other software developed at PCMDI (i.e., LATS and CDAT). Currently, cdunif can ingest five different uniform file formats: netCDF, HDF, DRS, VPOP and GrADS.

Compatibility with Current PCMDI Platforms (i.e., UNIX, Windows NT) – VCS was designed to run only on UNIX platforms. With new technology emerging from the internet and MicroSoft's operation system and new software, VCS must be able to operate in the current and impending future computer environments. That is, VCS must be able to run on Windows NT, UNIX, and via the Web. This effort will mean replacing the X/Motif interface with JAVA and replacing the X portion of the XGKS with JAVA (i.e., JAVAGKS or JGKS).

To meet the current and future needs of PCMDI and its global climate constituents, we have devised three possible scenarios for redesigning VCS. Each scenario meets the compatibility issues described above:

- a) Modify the existing VCS to run the CDAT input window from the VCS interface, thus, replacing the compute section in VCS. This scenario does not eliminate the above limitations. Once the modifications for CDAT in VCS are complete, we can redesign a new VCS to remove the limitations.
- b) Take parts of the X/Motif VCS interface and modify them for CDAT. Similar to what was done for the Colormap Editor and the Animation Panel. We will choose the most likely used GUIs (Graphical User's Interface) and modify them for CDAT accordingly. In this scenario, we freeze the current development of VCS until all possible GUIs have migrated over to CDAT. Once the GUI migration is complete, we redesign the new VCS with the necessary capabilities.
- c) We redesign the new VCS right away. Thus, telling the VCS users to wait for some extended period of time.

Note: this document does not discussed the way in which we will be re-implementing VCS. This topic is reserved for a later discussion.

Note: the priority order below is subject to the above discussion.

1) Assigned to Dean:

CDAT - VCS must be able to access CDAT for its calculation capabilities.

Person power required: ???

2) Assigned to Dean:

Attribute Changes for Computed Variables - When the user creates a computed variable, the variable should behave the same as any other variable retrieved from cdunif.

Person power required: ???

3) Assigned to ???:

GIF/JPEG Raster Output - VCS must be able to save gif or jpeg raster files for web output.

Person power required: ???

4) Assigned to ???:

CGM/MSWord - CGM output must be able to be imported into Microsoft Word.

Person power required: ???

5) Assigned to ???:

Postscript Files – VCS must produce postscript files directly like CGM and raster files.

Person power required: ???

6) Assigned to ???:

Fonts - Produce additional output fonts. Currently, VCS produce nine different fonts in the graphics display.

Person power required: ???

7) Assigned to ???:

Global Minimum and Maximum - Set global minimum and maximum for the graphics methods.

Person power required: ???

8) Assigned to ???:

CGM Hatch/Pattern Output - Fix hatch and pattern output to produce CGM files that mimic what is displayed on the VCS Canvas.

Person power required: ???

9) Assigned to ???:

Template Modification - Fix the template modification to allow for overall size change. That is, holding down the shift key and moving one of the template button (i.e., TL, TC, TR, etc.) will maintain the aspect ratio of the plot.

Person power required: ???

10) Assigned to ???:

Histogram Plot - VCS should be able to produce a histogram plot.

Person power required: ???

11) Assigned to ???:

Shaded Line Plot – VCS should be able to produce a shaded line plot.

Person power required: ???

12) Assigned to ???:

Portrait Plot – VCS should be able to produce portrait plots similar to GrADS.

Person power required: ???